Towards online structural state-estimation with sub-millisecond latency

Daniel Coble[1], Joud Satme[1], Ehsan Kabir[2], Austin R.J. Downey[1,3], Jason Bakos[4], David Andrews[2], Miaoqing Huang[2], Adrine Moura[5], and Jacob Dodson[6]

[1]Department of Mechanical Engineering, University of South Carolina, Columbia, SC 29208
[2]Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701
[3]Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC 29208
[4]Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208
[5] Applied Research Associates Emerald Coast Division 956 John Sims Pkwy W. Niceville, FL 32578
[6]Air Force Research Laboratory Munitions Directorate, Eglin Air Force Base, FL 32542

1 Introduction

High-rate structural state estimation is a field of focus for developing next-generation control schemes [1]. Orbital infrastructure, hypersonic vehicles, hard target penetrating munitions, and blast mitigation systems are intended to operate in high-rate dynamic environments [2–4]. Structures subjected to shock loads resulting in high accelerations of more than 100 Gs in less than 100 milliseconds are considered to be in the high-rate dynamic regime [5]. Due to the dynamic variations and high uncertainty of such environments, the estimated state of those systems should be updated within the sub-millisecond range. Minimizing the latency associated with state estimation algorithms allows for a faster response time, desirable for high-rate control applications. Utilizing the Dynamic Reproduction of Projectiles in Ballistic Environments for Advanced Research (DROPBEAR) system previously developed by the Air Force research laboratory and modeled by Joyce et al. [6], a coupled hardware-software system was designed to investigate the possibility of using a long short-term memory (LSTM) algorithm in a high-rate state estimation framework [7]. Specifically, this work presents a methodology of offline training with the DROPBEAR dataset and deployment to a real-time system where the immediate goal of this work is to feed the state estimator LSTM architecture the acceleration data with the prediction output being the pin location. Using data on the computation time of the real-time system, a timing model is developed to characterize models as real-time feasible or infeasible. Finally, a feasible model is selected for deployment on the real-time target. Experimental results and error are reported. The contributions of this work are a decrease in latency between state predictions and an investigation into the performance of models with different architectures.

2 Background

The DROPBEAR experimental testbed shown in figure 1 consists of a cantilevered beam fitted with an accelerometer, on the free end, to measure propagating vibrations [6]. A linear actuator mounted with movable pin support is used to alter the state of the structure simulating damage while a displacement sensor is used to measure the location of the pin movement. The continuously changing roller condition simulates the changing state of a structure as it accumulates damage. The data set used in this work is made available through a public repository [8].

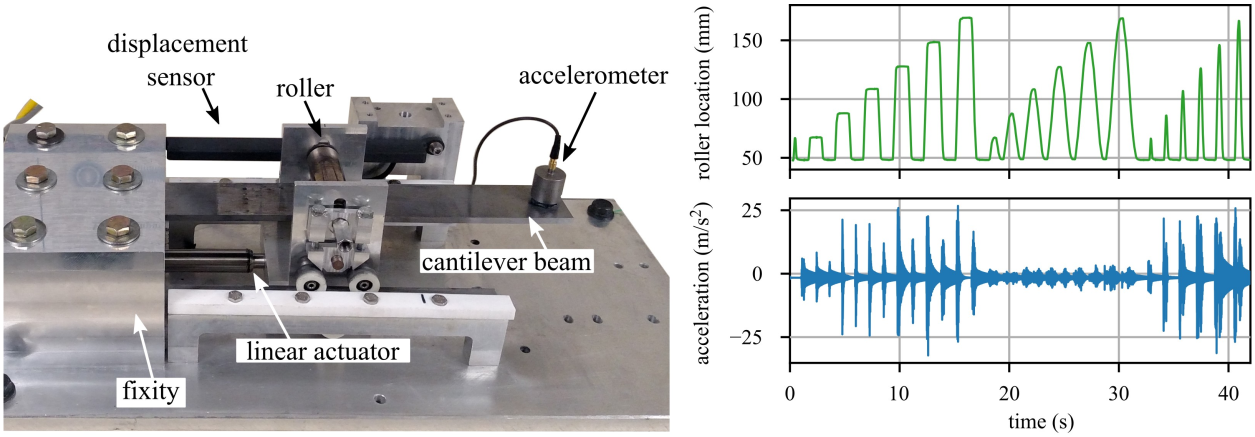

Figure 1: DROPBEAR experimental setup along with the displacement and acceleration signals [8] (Austin Downey, CC BY-SA 4.0).

Recurrent neural networks (RNNs), are a category of deep-learning architecture designed to handle time-series data. Long short-term memory (LSTM) models time-series data by augmenting state vectors h and c, which are trained to encode data about the system under study. Equations 1-6 describe a single-step forward pass of one LSTM layer.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{2}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{3}$$

$$\tilde{c}_t = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{4}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{5}$$

$$h_t = o_t \circ \tanh\left(c_t\right) \tag{6}$$

where the sigmoid activation function σ and hyperbolic tangent tanh are applied elementwise to the vector and $\circ$ is the elementwise, or Hadamard, product. The size of the state vectors, termed the units, determines complexity of patterns the LSTM is capable of representing. Alternatively, the complexity can be improved by increasing the number of LSTM layers in the model. In this work, models were trained using Tensorflow and Keras and implemented on the real-time target using LabVIEW Real-Time. The code for implementing LSTMs has been made publicly available through an open-source LabVIEW library [9].

3 Methodology

Figure 2 shows the form of models developed in this work. Acceleration is sampled across one timestep to be the input vector to the LSTM model and the state prediction is returned after one timestep of computation. Sampling and computation happen simultaneously at each timestep so that a new state prediction is returned every Δt. The timing requirement Δt determines the acceptable delay between the state occurrence and state prediction. For this paper, a timing requirement of 500 μs was selected.

To constrain the search space, a standard of 16 samples per timestep was chosen. A grid search was performed on the number model cells (between 1 to 3) and units per cell, (between 8 to 40 units), with each cell having the same number of units. Not all trained models were feasible for real-time deployment. Training was performed on sections of 0.1 s sampled randomly from the dataset. The state estimate after 0.1 s was used as the error source for backpropagation, and training ceased when error was detected to no

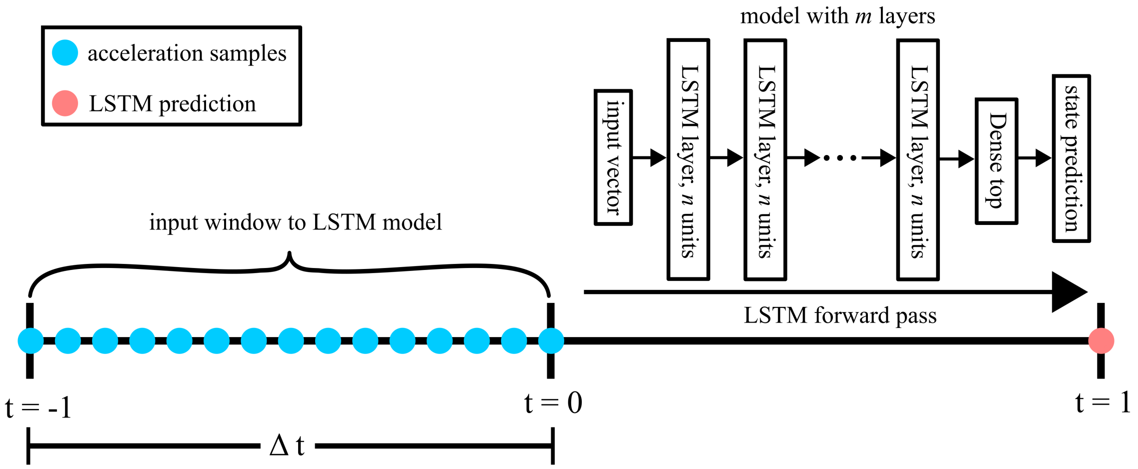

Figure 2: Timeline of sampling and computation for a single timestep of a LSTM forward pass.

longer be decreasing. This methodology, as compared to batch training on the entire dataset, performed better with respect to convergence time.

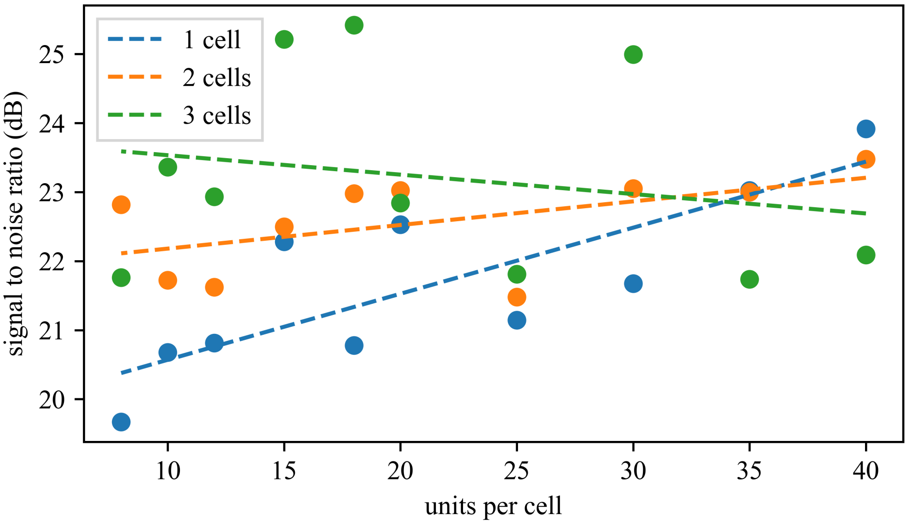

Figure 3: SNR_{dB} values of models with different numbers of cells and units.

Figure 3 shows the signal to noise ratio measured in decibels (SNR_{dB}) across the trained models. Models marked with dots are real-time feasible and models marked with crosses are not feasible. In general, SNR_{dB} increases with increasing units per cell, but this pattern did not hold for three-cell models, which exhibited a high variability in training with a downward slope. After performing the training search, the model with two cells and 15 units was selected for real-time edge implementation.

The test set-up shown in figure 4 consists of a host machine, data synthesis device, and real-time target machine. The data synthesis device, a cDAQ-9178 with a NI-9263 digital-to-analog module, reproduces the analog signal of the DROPBEAR dataset. Model calculations are run with an enforced timing requirement of 500 μs and the state prediction is then returned to the host machine via a first-in-first-out buffer. The real-time device used is a cRIO-9035 with a 1.33 GHz dual-core Intel Atom (E3825) manufactured by NI. Data is

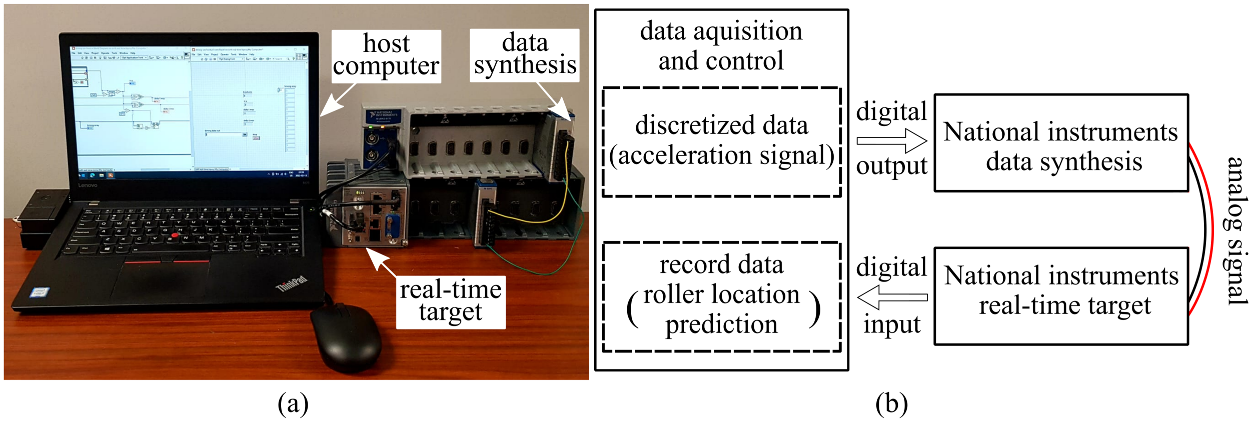
host computer
data synthesis
real-time target
data aquisition and control
discretized data (acceleration signal)
record data (roller location prediction)
digital output
digital input
National instruments data synthesis
National instruments real-time target
analog signal
(a)
(b)

Figure 4: Experimental setup of edge implementation along with a block diagram representation of the experiment.

resampled using a NI-9201 12-bit analog-to-digital module. The real-time target runs NI Linux Real-Time, an operating system developed by NI designed to improve determinism in the Linux kernel. LSTM models were developed in LabVIEW [9] and deployed to a LabVIEW Real-Time environment, which allows easy integration with data acquisition.

4 Results and Discussion

The real-time prediction of pin position is shown against the reference pin position in figure 5. Performance is reported in table 1 in terms of signal-to-noise ratio measured in decibels ($\mathrm{SNR_{dB}}$), root mean squared error (RMSE), and time response assurance criterion (TRAC), where TRAC is a metric varying between 0 to 1 measuring the correlation between two time series signals. The equation for TRAC is given in equation 7 for the reference time series vector y and prediction vector $\hat{y}$ [10].

$$\mathrm{TRAC} = \frac{(y^T \hat{y})^2}{(y^T y)(\hat{y}^T \hat{y})} \tag{7}$$

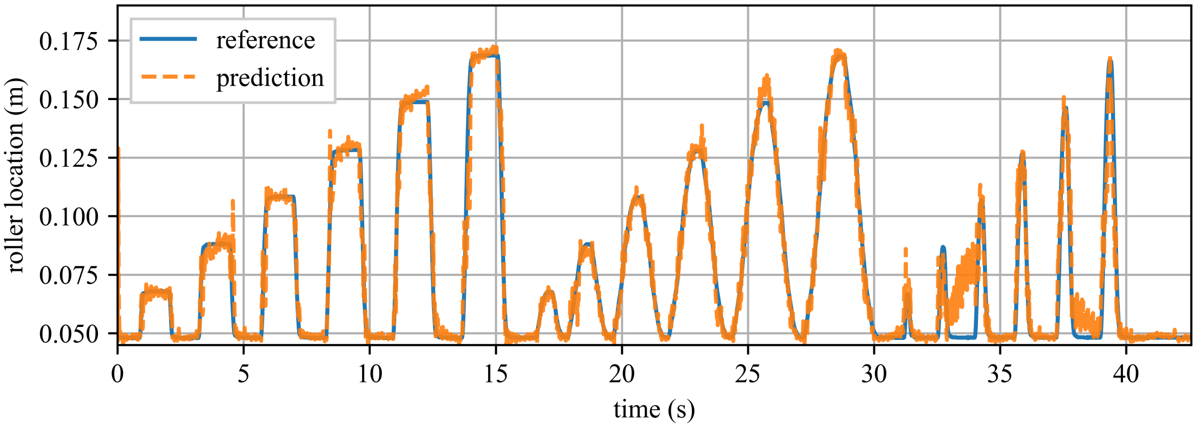

Figure 5: Results of the LSTM state estimation model on the real-time target.

Table 1: Error report with multiple metrics.

	simulated value	real-time value
$\mathrm{SNR}_{\mathrm{dB}}$	22.49	20.91
RMSE	6.44 mm	7.73 mm
TRAC	0.994	0.991

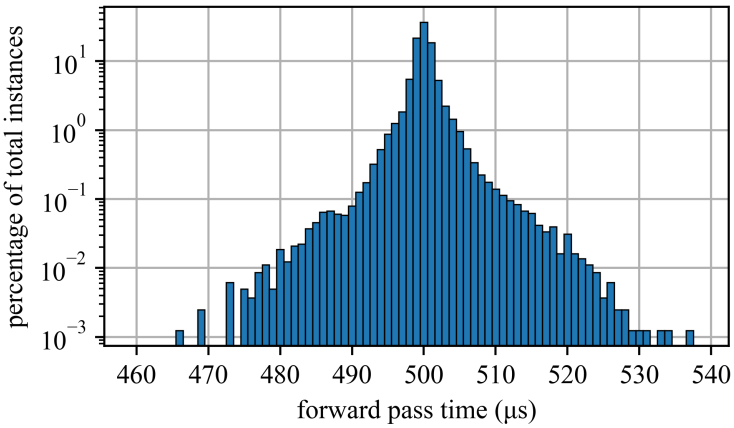

Figure 6: Timing distribution of LSTM single-timestep forward pass on real-time target.

The results reported for the simulated values are those from a computational pass across the dataset performed after training. The 1.59 dB loss of signal and 20.0% increase in RMSE error is a result of processes in signal regeneration and resampling. In addition to noise in generating and reading the signal, as signal generation and sampling occurred at different rates, the resampling which occurred digitally during training was performed via analog sampling during the experiment. Zero-order hold effects of the signal generation would then affect resampling, especially as the resampling rate (32 kS/s) is higher than the generation rate (25.6 kS/s).

Jitter, or deviation from the specified timing deadline, is a result of non-determinism introduced by the Linux kernel. To investigate the consistency of the real-time target in meeting the timing deadline, figure 6 shows the distribution of the forward pass timing. The distribution follows a normal distribution centered around the specified time of 500 μs with a standard deviation of 2.50 μs. The maximum overshoot, the longest time reported over the deadline, was 37 μs while the shortest time reported was 34 μs ahead of the deadline; resulting in a total jitter of 71 μs.

5 Conclusion

In this work, the problem of state estimation in real-time was addressed with the design of LSTM models and their implementation on real-time systems. An empirically-derived timing model was used to identify models capable of meeting a 500 μs deadline. A model was then verified with a deployment onto the real-time system. By comparing simulated and real-time results of the same model, error as the result of signal reproduction and resampling was quantified. An $\mathrm{SNR}_{\mathrm{dB}}$ of 20.91 was achieved on the real-time system. An investigation of the timing distribution of the real-time target demonstrated the 500 μs deadline with a standard deviation of 2.50 μs and maximum overshoot of 37 μs. The results show that LSTMs deployed to real-time systems are capable of achieving accurate state estimations at rates in the hundreds of microseconds. Future work will search for ways to further increase the rate of state estimation.

6 Acknowledgments

This material is based upon work partially supported by the Air Force Office of Scientific Research (AFOSR) through award no. FA9550-21-1-0083. This work is also partly supported by the National Science Foundation Grant numbers 1850012, 1937535, and 1956071. The support of these agencies is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the United States Air Force. (Distribution A. Approved for public release; distribution unlimited (AFRL-2023-1269)).

References Cited

[1] J. Dodson, A. Downey, S. Laflamme, M. D. Todd, A. G. Moura, Y. Wang, Z. Mao, P. Avitabile, and E. Blasch, "High-rate structural health monitoring and prognostics: An overview," in *Data Science in Engineering, Volume 9*. Springer International Publishing, oct 2021, pp. 213–217, [Online]. Available: http://dx.doi.org/10.1007/978-3-030-76004-5_23

[2] C. Stein, R. Roybal, P. Tlomak, and W. Wilson, "A review of hypervelocity debris testing at the air force research laboratory," *Space Debris*, vol. 2, no. 4, pp. 331–356, 2000. [Online]. Available: https://doi.org/10.1023/b:sdeb.0000030024.23336.f5

[3] R. P. Hallion, C. M. Bedke, and M. V. Schanz, *Hypersonic Weapons and US National Security: A 21st Century Breakthrough*. Mitchell Institute for Aerospace Studies, 2016. [Online]. Available: http://docs.wixstatic.com/ugd/a2dd91_b7016a5428ff42c8a21898ab9d0ec349.pdf

[4] H. Wadley, K. Dharmasena, M. He, R. McMeeking, A. Evans, T. Bui-Thanh, and R. Radovitzky, "An active concept for limiting injuries caused by air blasts," *International Journal of Impact Engineering*, vol. 37, no. 3, pp. 317–323, mar 2010. [Online]. Available: https://doi.org/10.1016/j.ijimpeng.2009.06.006

[5] J. Hong, S. Laflamme, J. Dodson, and B. Joyce, "Introduction to state estimation of high-rate system dynamics," *Sensors*, vol. 18, no. 2, p. 217, jan 2018, [Online]. Available: http://dx.doi.org/10.3390/s18010217

[6] B. Joyce, J. Dodson, S. Laflamme, and J. Hong, "An experimental test bed for developing high-rate structural health monitoring methods," *Shock and Vibration*, vol. 2018, pp. 1–10, jun 2018, [Online]. Available: http://dx.doi.org/10.1155/2018/3827463

[7] J. Satme, D. Coble, B. Priddy, A. R. Downey, J. D. Bakos, and G. Comert, "Progress towards data-driven high-rate structural state estimation on edge computing devices," in *Proceedings of the ASME 2022 International Design Engineering Technical Conferences*, 2022.

[8] A. Downey, J. Hong, J. Dodson, M. Carroll, and J. Scheppegrell, "Dataset-2-dropbear-acceleration-vs-roller-displacement," Apr. 2020. [Online]. Available: https://github.com/High-Rate-SHM-Working-Group/Dataset-2-DROPBEAR-Acceleration-vs-Roller-Displacement

[9] D. Coble and A. Downey, "LabVIEW-LSTM," GitHub, 2022. [Online]. Available: https://github.com/ARTS-Laboratory/LabVIEW-LSTM

[10] P. Avitabile and P. Pingle, “Prediction of full field dynamic strain from limited sets of measured data,” *Shock and vibration*, vol. 19, no. 5, pp. 765–785, 2012.